Assignment 2

Due date: Feb 28, 2013

General instructions:

- Submit your solutions by typesetting in LaTeX.
- Write your solutions by furnishing all relevant details (you may assume the results already covered in the class or previous homework problems).
- You are strongly urged to solve the problems by yourself.
- If you discuss with someone else or refer to any material (other than the course notes) then please put a reference in your answer script stating clearly whom or what you have consulted with and how it has benifited you. We would appreciate your honesty.
- If you need any clarification, please ask the instructor.

Total: 40 points

In the following problems, $\mathbb{N}$ is the set of natural numbers and $\mathbb{F}_q$ is the finite field with q elements.

1. (**5 points**) A number n is a Carmichael number if n is composite and for every $a \in \mathbb{N}$ that is coprime to n, $a^{n-1} = 1 \mod n$. (Recall from class that these are the numbers that fail Fermat's primality test).

   (a) (**3 points**) Show that a Carmichael number is square-free (i.e. there is no prime p such that p^2 divides n).

   (b) (**2 points**) Prove that a number n is a Carmichael number if and only if n is square-free and for every prime factor p of n, $p-1$ divides $n-1$.

2. (**6 points**) Suppose $n = pq$, where p and q are primes and let $(\mathbb{Z}_n^\times, \cdot)$ denote the group of numbers in $\{1, \ldots, n-1\}$ that are coprime to n under multiplication modulo n. In this exercise, you will show that the problem of computing the order of an element in $(\mathbb{Z}_n^\times, \cdot)$ is polynomial time equivalent to the problem of factoring integers (modulo the use of randomness). Prove the following claims.

   (a) (**4 points**) Given n, if there is a deterministic algorithm to compute the order of an element in $(\mathbb{Z}_n^\times, \cdot)$ in $(\log n)^{O(1)}$ time, then there is a randomized algorithm to factor n also in $(\log n)^{O(1)}$ time.

(b) (**2 points**) If there is a deterministic algorithm to factor integers in polynomial time then there is a deterministic algorithm to compute the order of an element in $(\mathbb{Z}_n^\times, \cdot)$ in $(\log n)^{O(1)}$ time.

3. (**5 points**) Let $n \in \mathbb{N}_{\geq 1}$.

   (a) (**1 point**) Prove that the following are equivalent for $u \in \mathbb{Z}_n$:
   - $u \in \mathbb{Z}_n^\times$ and order of u in $\mathbb{Z}_n^\times$ is $n-1$.
   - $u^{n-1} = 1$ and $u^{(n-1)/p} \neq 1 \mod n$ for all prime divisors p of $n-1$.

   We will call an $u \in \mathbb{Z}_n$ with these properties a 'witness' for the primality of n.

   (b) (**1 point**) Prove that n is a prime if and only if it has a witness.

   (c) (**3 points**) A 'certificate' C for the primality of n is defined recursively as follows:
   - $C = (2, 1)$ for $n = 2$.
   - $C = (n, u; p_1, e_1, \ldots, p_r, e_r; C_1, \ldots, C_r)$ if $n \geq 3$ such that
     - u is a 'witness' for n,
     - $p_1 < \ldots < p_r \in \mathbb{N}_{\geq 2}$ are primes, $e_1, \ldots, e_r \in \mathbb{N}_{\geq 0}$, and $n - 1 = p_1^{e_1} \ldots p_r^{e_r}$ is the prime factorization of $n-1$.
     - For all i, C_i is a 'certificate' for the primality of p_i.

   Prove that the length of a 'certificate' of n is $(\log n)^{O(1)}$ and given a certificate of n we can check its correctness in $(\log n)^{O(1)}$ time. Infer that PRIMES $\in$ NP.

4. (**10 points**) In this exercise, we will see how to compute the square root of an arbitrary element in a prime field $\mathbb{F}_p$ in $(\log p)^{O(1)}$ time, starting with any quadratic non-residue in $\mathbb{F}_p$. (Recall that an element x in $\mathbb{F}_p$ is a quadratic non-residue if there is no $y \in \mathbb{F}_p$ such that $y^2 = x$.) Let $a \in \mathbb{F}_p$ be an input element whose square root we wish to compute. Assume that we know η, an arbitrarily fixed quadratic non-residue in $\mathbb{F}_p$.

   (a) (**1 point**) Give a test, running in time $(\log p)^{O(1)}$, to confirm that a is actually a quadratic residue.

   (b) (**3 points**) Suppose $p - 1 = 2^t w$, where w is odd. Show that if we can compute the square root of a^w in time T then we can also compute the square root of a in time $T + (\log p)^{O(1)}$.

   (c) (**2 points**) Let $a' = a^w$. It follows from (b) that it is sufficient to compute the square root of a' efficiently. Notice that the element a' belongs to $G_w = \{c^w : c \in \mathbb{F}_p\}$, the subgroup of $\mathbb{F}_p^\times$ of size 2^t. Prove that $\eta' \stackrel{def}{=} \eta^w$ is a generator of the cyclic group G_w.

   (d) (**4 points**) Infer from (c) that there is an even number $e < 2^t$ such that $\eta'^e = a'$. Therefore, if we can find e then $\eta'^{e/2}$ is a square root of a'. Show that e can be computed in $(\log p)^{O(1)}$ time.

5. (**4 points**) Let $\mathcal{R}$ be a commutative ring (with unity), $k \in \mathbb{N}_{>0}$, and $f, g \in \mathcal{R}[x]$ with $f(0) = 1$ and $fg = 1 \mod x^k$.

   (a) (**1 point**) Let $d \in \mathbb{N}$, $e = 1 - fg$, and $h = g \cdot (e^{d-1} + e^{d-2} + \ldots + e + 1)$. Prove that $fh = 1 \mod x^{dk}$.

   (b) (**3 points**) State an algorithm for computing inverse of f modulo x^ℓ, where ℓ is a power of d and analyze its time complexity.

6. (**10 points**) *Background:* In the class, we have discussed a randomized procedure to find an irreducible polynomial over a finite field. In this exercise, we will see how to find irreducible polynomials deterministically for a special case. Let $\mathbb{F}_p$ be a prime field and $q_1, \ldots, q_\ell$ the distinct prime factors of $p-1$.

   Goal: To find an irreducible polynomial over $\mathbb{F}_p$ of degree $n = q_1 q_2 \cdots q_\ell$.

   Your task is to prove the following.

   (a) (**4 points**) Let a_i be a q_i-th power non-residue in $\mathbb{F}_p$. Prove that $x^{q_i} - a_i$ is irreducible over $\mathbb{F}_p$.

   (For your information: The Extended Riemann Hypothesis (ERH) implies that the value of the least q_i-th power non-residue in $\mathbb{F}_p$ is bounded by $O((\log p)^2)$ - so it is easy to find.)

   (b) (**6 points**) If we have irreducible polynomials $f, g \in \mathbb{F}_p$ of degree m and k, respectively, where m is relatively prime to k, then we can find an irreducible polynomial of degree mk over $\mathbb{F}_p$.

   Infer from (a) and (b) that we can find an irreducible polynomial of degree n deterministically (putting our faith in the ERH) in time polynomial in n and $\log p$.

   To help you appreciate the usefulness of this process, here's an example case: Start with the finite field $\mathbb{F}_7$ and construct an irreducible polynomial of degree $7 - 1 = 2 \cdot 3 = 6$ over $\mathbb{F}_7$. With this construct and work with the extended field $\mathbb{F}_{7^6} = \mathbb{F}_{117649}$!